

# fifo\_ctrl.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2021/06/29

---

## INFORMATION

---

### Brief

---

Control block for fifo operations, emulates xilinx fifo.

### License MIT

---

Copyright 2021 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## fifo\_ctrl

---

```
module fifo_ctrl #(
    parameter
    FIFO_DEPTH
    =
    256,
    parameter
    BYTE_WIDTH
    =
    1,
    parameter
    ADDR_WIDTH
```

```

    =
    1,
    parameter
    COUNT_WIDTH
    =
    1,
    parameter
    GREY_CODE
    =
    1,
    parameter
    COUNT_DELAY
    =
    1,
    parameter
    COUNT_ENA
    =
    1,
    parameter
    ACK_ENA
    =
    0,
    parameter
    FWFT
    =
    0
) ( input rd_clk, input rd_rstn, input rd_en, output [ADDR_WIDTH-1:0] rd_addr

```

Control block for fifo operations, emulates xilinx fifo.

## Parameters

<b>FIFO_DEPTH</b> parameter	Depth of the fifo, must be a power of two number(divisable aka $256 = 2^8$ ). Any non-power of two will be rounded up to the next closest.
<b>BYTE_WIDTH</b> parameter	How many bytes wide the data in/out will be.
<b>ADDR_WIDTH</b> parameter	Width of the RAM address bus to write data to.
<b>COUNT_WIDTH</b> parameter	Data count output width in bits. Should be the same power of two as fifo depth(256 for fifo depth... this should be 8).
<b>GREY_CODE</b> parameter	RAM address uses grey code instead of linear addressing.
<b>COUNT_DELAY</b> parameter	Delay count by one clock cycle of the data count clock. Set this to 0 to disable (only disable if read/write/data_count are on the same clock domain!).
<b>COUNT_ENA</b> parameter	Enable the count output.
<b>ACK_ENA</b> parameter	Enable ack on write.
<b>FWFT</b> parameter	1 for first word fall through mode. 0 for normal.

## Ports

<b>rd_clk</b>	Clock for read data
<b>rd_rstn</b>	Negative edge reset for read.
<b>rd_en</b>	Active high enable of read interface.
<b>rd_addr</b>	Address to read data from in RAM.

<b>rd_valid</b>	Active high output that the data is valid.
<b>rd_mem_en</b>	Active high enable to read from RAM.
<b>rd_empty</b>	Active high output when read is empty.
<b>wr_clk</b>	Clock for write data
<b>wr_rstn</b>	Negative edge reset for write
<b>wr_en</b>	Active high enable of write interface.
<b>wr_addr</b>	Address to write data to in RAM.
<b>wr_ack</b>	Active high when enabled, that data write has been done.
<b>wr_mem_en</b>	Active high enable to write to RAM.
<b>wr_full</b>	Active high output that the FIFO is full.
<b>data_count_clk</b>	Clock for data count
<b>data_count_rstn</b>	Negative edge reset for data count.
<b>data_count</b>	Output that indicates the amount of data in the FIFO.